Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently Amended) A method for managing a data imaging service from a management terminal in a distributed computer system having a host computer system with at least one storage device connected to the computer system by driver software, the method comprising:

    (a)   inserting an interface layer between the driver software and the storage device, the interface layer exporting a platform dependent API <u>comprising a plurality of API methods that can be used to control</u> ~~and controlling~~ data passing between the driver software and the storage device;

    (b)   running, in the host computer system, management facade software that ~~converts the interface layer API to~~ <u>receives calls to</u> platform-independent ~~method calls~~ <u>methods and generates at least one API method call to the interface layer in order to execute the platform-independent method calls</u>;

    (c)   running, in the host computer system, a federated bean that generates <u>platform-independent</u> method calls to the management facade to control the interface layer <u>via the plurality of API methods</u>; and

    (d)   controlling the federated bean to designate master volumes, shadow volumes and bitmap volumes and to transfer data between specified master and shadow volumes.

2.   (Original) The method of claim 1 wherein step (d) comprises controlling the federated bean with a command line interface.

3. (Original) The method of claim 1 wherein step (d) comprises controlling the federated bean with a graphical user interface.

4. (Original) The method of claim 1 wherein step (d) comprises:
(d1) creating a volume set; and
(d2) designating a master volume, a shadow volume and a bitmap volume as part of the volume set; and
(d3) performing data imaging operations on the volume set.

5. (Original) The method of claim 4 wherein a plurality of volume sets are created and wherein the method further comprises:
(e) creating a set group; and
(f) adding selected volume sets to the set group; and
(g) controlling the set group with a single command to perform data imaging operations on each set in the set group.

6. (Original) The method of claim 4 further comprising attaching an overflow volume to the volume set.

7. (Original) The method of claim 4 wherein the computer system has a first host with a volume set thereon and a second host and the method comprises exporting a shadow volume in the volume set from the first host.

8. (Original) The method of claim 7 further comprising importing the shadow volume exported by the first host into the second host.

9. (Currently Amended) Apparatus for managing a data imaging service from a management terminal in a distributed computer system having a host computer system with at least one storage device connected to the computer system by driver software, the apparatus comprising:

3

5   an interface layer located between the driver software and the storage

6   device, the interface layer exporting a platform dependent API comprising a

7   plurality of API methods that can be used to control ~~and controlling~~ data passing

8   between the driver software and the storage device;

9   management facade software that runs in the host computer system and

10   ~~converts the interface layer API to~~ receives calls to platform-independent ~~method~~

11   ~~calls~~ methods and generates at least one API method call to the interface layer in

12   order to execute the platform-independent method calls;

13   a federated bean that runs in the host computer system and generates

14   platform-independent method calls to the management facade to control the

15   interface layer via the plurality of API methods; and

16   a presentation program that controls the federated bean to designate

17   master volumes, shadow volumes and bitmap volumes and to transfer data

18   between specified master and shadow volumes.

1   10.   (Original) The apparatus of claim 9 wherein the presentation program comprises

2   a command line interface.

1   11.   (Original) The apparatus of claim 9 wherein the presentation program comprises

2   a graphical user interface.

1   12.   (Original) The apparatus of claim 9 wherein the presentation program comprises:

2   program methods for creating a volume set; and

3   a screen display for designating a master volume, a shadow volume and a

4   bitmap volume as part of the volume set; and

5   program methods for performing data imaging operations on the volume

6   set.

1   13.   (Original) The apparatus of claim 12 wherein a plurality of volume sets are

2   created and wherein the apparatus further comprises:

4

3   program methods for creating a set group; and

4   a screen display for adding selected volume sets to the set group; and

5   program methods for controlling the set group with a single command to

6   perform data imaging operations on each set in the set group.

1   14.   (Original) The apparatus of claim 12 further comprising program methods for

2   attaching an overflow volume to the volume set.

1   15.   (Original) The apparatus of claim 12 wherein the computer system has a first

2   host with a volume set thereon and a second host and the apparatus comprises

3   means for exporting a shadow volume in the volume set from the first host.

1   16.   (Original) The apparatus of claim 15 further comprising means for importing the

2   shadow volume exported by the first host into the second host.

1   17.   (Currently Amended) A computer program product for managing a data imaging

2   service from a management terminal in a distributed computer system having a

3   host computer system with at least one storage device connected to the

4   computer system by driver software, the computer program product comprising a

5   computer usable medium having computer readable program code thereon,

6   including:

7   interface layer program code located between the driver software and the

8   storage device, the interface layer code exporting a platform dependent API

9   comprising a plurality of API methods that can be called to control ~~and controlling~~

10   data passing between the driver software and the storage device;

11   management facade software that runs in the host computer system and

12   ~~converts the interface layer API to~~ receives calls to platform-independent ~~method~~

13   ~~calls~~ methods and generates at least one API method call to the interface layer in

14   order to execute the platform-independent method calls;

5

15   a federated bean that runs in the host computer system and generates

16   platform-independent method calls to the management facade to control the

17   interface layer via the plurality of API methods; and

18   a presentation program that controls the federated bean to designate

19   master volumes, shadow volumes and bitmap volumes and to transfer data

20   between specified master and shadow volumes.

1   18.   (Original) The computer program product of claim 17 wherein the presentation

2   program comprises a command line interface.

1   19.   (Original) The computer program product of claim 17 wherein the presentation

2   program comprises a graphical user interface.

1   20.   (Currently Amended) A computer data signal embodied in a carrier wave for

2   managing a data imaging service from a management terminal in a distributed

3   computer system having a host computer system with at least one storage

4   device connected to the computer system by driver software, the computer data

5   signal comprising:

6   interface layer program code located between the driver software and the

7   storage device, the interface layer code exporting a platform dependent API

8   comprising a plurality of API methods that can be called to control and controlling

9   data passing between the driver software and the storage device;

10   management facade software that runs in the host computer system and

11   converts the interface layer API to receives calls to platform-independent method

12   calls methods and generates at least one API method call to the interface layer in

13   order to execute the platform-independent method calls;

14   a federated bean that runs in the host computer system and generates

15   platform-independent method calls to the management facade to control the

16   interface layer via the plurality of API methods; and

6

17   a presentation program that controls the federated bean to designate

18 master volumes, shadow volumes and bitmap volumes and to transfer data

19 between specified master and shadow volumes.